Some Aspects of Optical Devices and Protocols in Scientific Parallel Computing

Final Synopsis of the PhD work to be submitted to

Department of Post-Graduate Studies and Research in Electronics

Mangalore University

by

Mohammed Mahfooz Sheikh

Under the Supervision of

Dr. A. M. Khan

Associate Professor and Chairman,
Department of Post-Graduate Studies and Research in Electronics,
Mangalore University, Mangalagangotri-574 199,
Karnataka, India.

and

Dr. U N Sinha
(co – guide)
Distinguished Scientist
CSIR C-MMACS,
Belur Campus, Bangalore – 560037,
Karnataka, India.

September 2012

INTRODUCTION

Large scale computing has played important role in solving complex scientific and technological problems. Without large scale computational resources genome sequencing could not have been possible (Mark Delderfield *et al* 2008). Without the help from primitive computer, design of atom bomb would not have been feasible (Herman Goldstine H & John Von Nuemann 1976). New drug development routinely uses large scale computing (Hausheer F H 1992). Many new discoveries have been result of large scale computations. For example, solitary waves were found by Ulam and his colleagues using large scale computing (Fermi E *et al* 1955); space missions demand massive computing for re-entry trajectories of space vehicles and numerical precision exceeding 20 digits are quite common. It is, therefore, not surprising that requirement of large scale computations has led to development of parallel machines with history dating back to 1960's (Jon Squire S & Sandra Palais M 1963; Koczela L J & Wang G Y 1969). The story of developments of the computers in use till early 70's is well documented and vividly presented in the references (Akira Kasahara 1970; Herman Goldstine H 1973; Charles Eames & Ray Eames 1973; David Kuck J 1978; Presper Eckart Jr J 1980; John Mauchly W 1980) which makes a fascinating reading. The main interest here is to get a feel of the growth rate of power of computing. The diagram in the Figure 1 is taken from Michael Quinn J (1988). It is a typical figure which is found frequently in the related writings that shows schematically the history of growth of computing power since 1950. For further growth it is not surprising that route of parallel processing emerged in a natural way as the switching time and propagation delay were being limited with the limit set by the velocity of light.

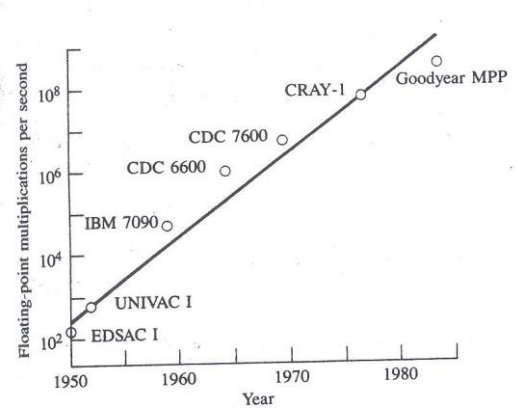

Fig 1. Growth rate of computing power since 1950, courtesy Michael Quinn J (1988).

In parallel processing the task is divided among various Processing Elements (PEs) that execute the jobs in parallel. It is implicitly assumed here that the task is amenable to parallel processing and communication mechanism is in place so that PEs may work on the subtasks of the main task and yet complete the main task as if the process is carried out on a single virtual sequential computing machine. Communication paradigm appears at a cross road at this point. Communication issues in the earlier sequential computer were related to interconnection of CPU elements, for example the design of CDC 7600 was related to enabling the CPU for floating point division in a single cycle or in modern times making a super pipeline structure of Pentium for doing a single floating point operation per cycle in a pipeline; whereas for parallel processing communication it normally refers to communication among PEs on a macroscopic or gross or at course grained level.

INDIAN CONTRIBUTION TO COMMUNICATION TECHNOLOGY

When seen in the background of the international efforts, Indian efforts offer a portrayal which is not one of following but that one of competing. National Aerospace Laboratories (NAL), Bangalore has been engaged in Computational Fluid Dynamics (CFD) calculations since beginning needing large computational resources. Inadequacy of computational resources had led to the initiation of Flosolver programme in 1986 to overcome this problem through the route of computing by Prof. R. Narasimha, then director of NAL. Indian success in the parallel computing started with the development of Flosolvers Mk1 to cater the requirements of applications in Computational Fluid Dynamics (Sinha U N *et al* 1988).

The success of NAL's parallel computing resulted in setting up of CDAC, Centre for Development of Advanced Computing in 1988, a national initiative on a much larger and ambitious scale (Bhatkar V P 1991). CDAC completed its PARAM parallel computer in July 1991 with a peak performance of 1 GFLOPS. It was a 256-node system comprising of a group of 64 nodes arranged in the form of computing clusters. These computing clusters were interconnected internally and externally by means of cross-point switches to form a 256-node system. PARAM Yuva being the latest of the PARAM series of supercomputer developed in June 2009. It is capable of performing at 38.1 TFLOPS. The

hardware has been supported by Hewlett-Packard with HP cluster platform 3000. It consists of 4608 Intel Xeon processors interconnected using the Infiniband technology. The major applications of PARAM are in long-range weather forecasting, remote sensing, drug design, molecular modeling etc.

Defence Research and Development Organisation (DRDO) completed its PACE (Processor for Aerodynamic Computations and Evaluation) parallel computer in 1991 housing a 128 Processing Element (PE) system, with each PEs interconnected in the form of hypercube architecture (Neelakantan K *et al* 1990).

The CSIR Centre for Mathematical Modelling and Computer Simulations possess a parallel computer SGI Altix ICE8400 EX that consists of 96 nodes of 12 cores per node interconnected by means of InfiniBand Interconnect using an hypercube architecture. It consists of a storage capacity of 30 Tbytes. The centre also posseses SGI Altix 4700 Compute server consisting of 288 cores inter connected by means of 4Gbps fibre channel based Gigabit Ethernet network.

EKA was built by the Computational Research Laboratories with technical assistance and hardware provided by Hewlett-Packard. It was installed in November 2007, and then it was the 4th fastest in the world and fastest in Asia. EKA has 1,794 computing nodes and has a theoretical peak performance of 172.2 Teraflops.

SAGA-220 was built by the Indian Space Research Organization (ISRO) in 2011. The system uses 400 NVIDIA Tesla 2070 GPUs and 400 Intel Quad Core Xeon CPUs connected with each other using a high speed interconnect supplied by WIPRO. Each NVIDIA Tesla 2070 GPUs is capable of delivering 515 gigaflops compared to the Xeon CPU's more modest contribution of 50 gigaflops. Its peak rating is 220 TFLOPS.

The Floswitch in Flosolver series of parallel computers has been an outcome of experience gained in designing parallel machines having better scalability features for tightly coupled class of problems. It emerged that the paradigm of only message communication limits the performance. If switches were given the extra feature of message processing, the scalability could be significantly enhanced. This was the genesis of Floswitch. In other words, this paradigm of communication switch processes information while communicating and it results in a new architecture which has proven very effective for problem like meteorological computing, DNS computing, panel

techniques for aircraft wing load calculation and many other problems of this class. The effectiveness of this approach can be easily accessed by super linear speed up of Navier Stokes calculation which is something like a milestone in judging effectiveness of parallel computing (Venkatesh T N *et al* 2005).

As the number of CPUs or motherboards linked to a single Floswitch is limited by the constraints of the techniques in PCB fabrication technology, optical routes were adopted for increasing the interconnections. Flosolver Unit of National Aerospace Laboratories in Bangalore has an appreciable work on development of scientific parallel computer under progress. This provided the platform for the proposed work to be carried out. The study and enhancement of the Floswitch was the concern of study in the proposed thesis.

OBJECTIVES OF THE PROPOSED WORK

1. To perform a detailed study on the various aspects of Scientific Parallel Computing with an emphasis on the type of communication technology used.
2. To carry out the study on the Indian contributions to the communication technology in parallel computing and birth of Floswitch.
3. To determine the limitations in the existing Floswitch configuration and develop avenues for improvement evolving into a new design of Floswitch.
4. To gradually develop efficient message processing algorithm for the new design of Floswitch.
5. To conduct a qualitative analysis in order to substantiate the outcome of new improved message processing algorithm.

LIMITATIONS IN THE EXISTING FLOSWITCH CONFIGURATION

Processing of one packet data (1024 dwords) in the existing Floswitch configuration as shown in Figure 2, involves the following steps:

1. One read clock is required in order to read data from all the four memories of the respective PEs to the input buffer. Since all the PEs have different memory blocks, these read are performed in parallel. Thus, 1024 dwords read will take 1024 clocks.

2. The computation of input data from all four PEs in the best possible case is 3 clocks. Here, 1024 dwords from each PE requires 3 x 1024 = 3072 clocks.

3. Lastly, one write clock is required to write each resultant dword from the output BRAM to the respective memories of all the respective PEs in broadcast. Thus, 1024 dwords write requires 1024 clocks.

Therefore, the complete operation for a packet of data consisting of 1024 dwords requires 1024 + 3072 + 1024 = **5120 clocks**.

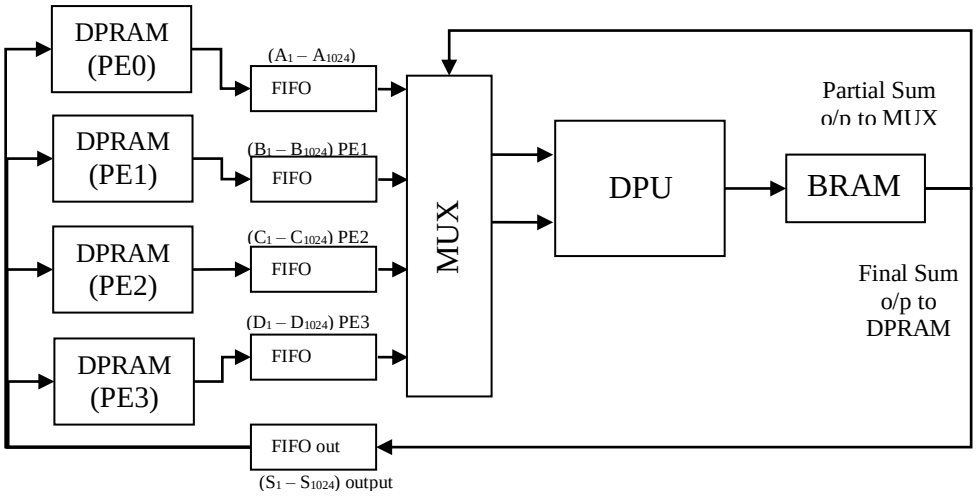

Fig 2. Block diagram of the Data Processing Engine in the Floswitch.

DESIGN OF NEW MESSAGE PROCESSING ALGORITHM

The clock descriptions for the float add operations using the proposed Floswitch architecture are as shown in the Table 1 below:

Clock no.	Activity during the clock			Store result in memory
	Read data	Float add operation	Output	
1	a_1, b_1 c_1, d_1	Adder1: idle Adder2: idle Adder3: idle	None	None
2	a_2, b_2 c_2, d_2	Adder1: $p_1 = a_1 + b_1$ Adder2: $q_1 = c_1 + d_1$ Adder3: idle	None	None

3	a_3, b_3 c_3, d_3	Adder1: $p_2 = a_2 + b_2$ Adder2: $q_2 = c_2 + d_2$ Adder3: idle	None	None
4	a_4, b_4 c_4, d_4	Adder1: $p_3 = a_3 + b_3$ Adder2: $q_3 = c_3 + d_3$ Adder3: $s_1 = p_1 + q_1$	None	None
5	a_5, b_5 c_5, d_5	Adder1: $p_4 = a_4 + b_4$ Adder2: $q_4 = c_4 + d_4$ Adder3: $s_2 = p_2 + q_2$	s_1	None
6	a_6, b_6 c_6, d_6	Adder1: $p_5 = a_5 + b_5$ Adder2: $q_5 = c_5 + d_5$ Adder3: $s_3 = p_3 + q_3$	s_2	s_1
7	a_7, b_7 c_7, d_7	Adder1: $p_6 = a_6 + b_6$ Adder2: $q_6 = c_6 + d_6$ Adder3: $s_4 = p_4 + q_4$	s_3	s_2
8	a_8, b_8 c_8, d_8	Adder1: $p_7 = a_7 + b_7$ Adder2: $q_7 = c_7 + d_7$ Adder3: $s_5 = p_5 + q_5$	s_4	s_3
9	None	Adder1: $p_8 = a_8 + b_8$ Adder2: $q_8 = c_8 + d_8$ Adder3: $s_6 = p_6 + q_6$	s_5	s_4
10	None	Adder1: idle Adder2: idle Adder3: $s_7 = p_7 + q_7$	s_6	s_5
11	None	Adder1: idle Adder2: idle Adder3: $s_8 = p_8 + q_8$	s_7	s_6
12	None	Adder1: idle Adder2: idle Adder3: idle	s_8	s_7
13	None	Adder1: idle Adder2: idle Adder3: idle	None	s_8

Table 1. Description of Float add operation for 8 dwords packet.

The proposed Floswitch DPU is as shown in Figure 3a & 3b. As evident from the Table 1, where the computation of a packet of data consisting of 8 dwords is considered for simplicity, an overhead of 3 clocks is clearly understood from the first three rows in the 'output' column. Hence the computation of 1024 dwords packet would also involve an overhead of 3 clocks only. Which means the complete operation requires 1024 + 3 =

1027 clocks for the computation of 1 packet data (1024 dwords) as compared to the existing Floswitch architecture where previous calculations have shown **5120 clocks**.

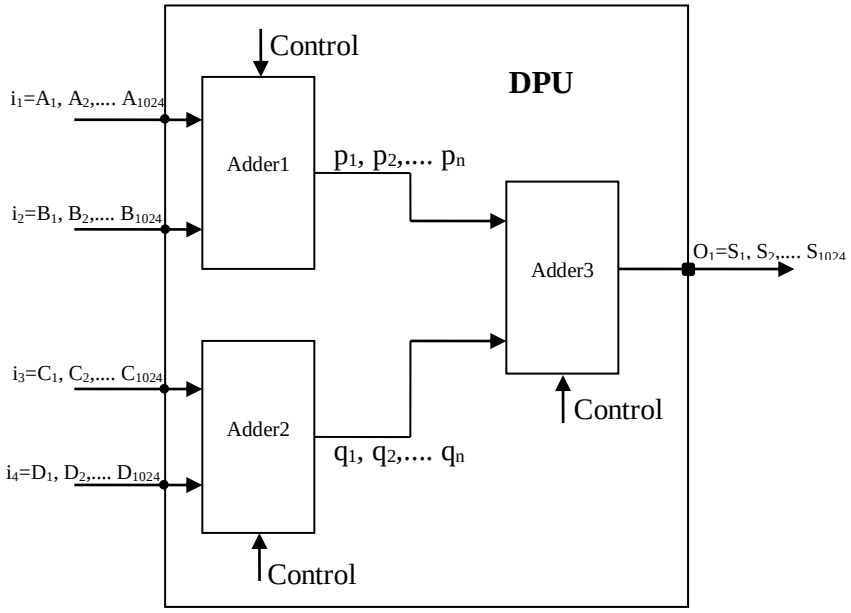

Fig 3a. Block diagram of the new DPU architecture.

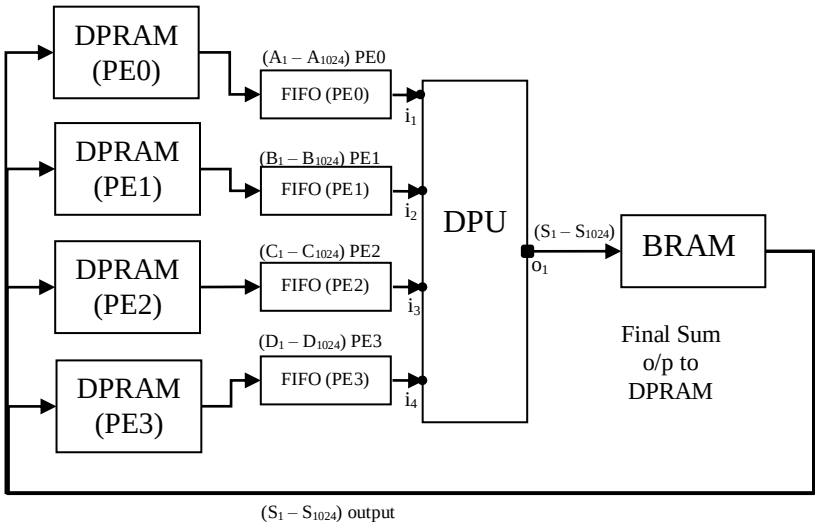

Fig 3b. Block diagram of the new DPU architecture implemented in the existing Data Processing Engine.

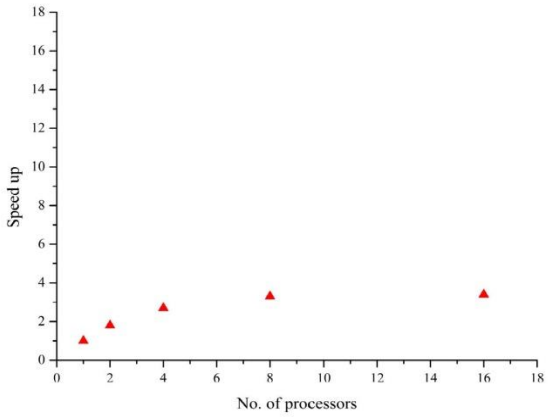

Fig. 4. Efficiency of VARSHA model for various number of processors. (using MPI communication protocol, Ethernet 1GB rating)

TIMING ANALYSES FOR THE LEGACY CODE VARSHA

The VARSHA code is required to be profiled for only a specific segment of the code which is compute intensive. Moreover, the complete profiling of VARSHA will generate huge amount of data. With all these computations in mind, the present study utilizes fourth time step of the forecast for profiling. During each of the time steps, the code computes, Fourier Transform of 256 latitudes of 512 points each for 18 atmospheric levels. It also computes Legendre Transform for equal number of points and also computes the nonlinear part of dynamical calculations. These computations in the fourth time step and their communication to other boards depending on the number of boards are profiled and presented in Table 2. The speedup trend is as shown in the previous graph in Figure 4. Also the bandwidth analysis techniques for the speedup trend in Figure 4 are explained as follows:

Boards	CPU Processing Time (msec)	Communication Time (msec)	Actual Processing Time (msec)	Speed up
1	3077	2	3079	1
2	1541	181	1722	1.8
4	774	352	1126	2.7
8	392	528	920	3.3
16	201	700	901	3.4

Table 2: Speed up of Original Application code.

It is evident from Table 2 that the speedup is not appreciable beyond 8 processors. But if the bandwidth can be improved, i.e. the communication time will be reduced; there will be increase in speedup for more number of processors. In other words, if t is the actual processing time then

$$t = t_{cpu} + t_{comm} \qquad (1)$$

where t_{cpu} is the computation time and t_{comm} is the communication time. The bandwidth being increased by a factor of k, the actual or effective processing time then computed would be

$$t = t_{cpu} + \frac{t_{comm}}{k} \text{ ; where k is a positive integer} \qquad (2)$$

Here, as k is inversely proportional to the actual processing time t, any increase in k value shall tend to minimise the time t and increase the speedup as evident from the Table 3.

The table 4 gives the speedup values for different bandwidth of the order $k = 2^m$ where $0 \leq m \leq 4$. The graph in Figure 5 shows the change in speedup for various bandwidths. It is observed that as the bandwidth scaling increases i.e. the communication time reduces the speedup of the application code increases when more number of processors is used. Thus, the reduction in communication time with the scaling of bandwidth significantly decides the computational efficiency of the legacy code VARSHA.

Boards	**CPU Processing Time (msec)** t_{cpu}	**Communication Time (msec)** t_{comm}/k $(k=8)$	**Actual Processing Time (msec)** t_p	**Speed up**
1	**3077**	**0.3**	**3077.4**	**1**
2	**1541**	**22.6**	**1577.2**	**2**
4	**774**	**44.0**	**814.4**	**3.8**
8	**392**	**66.0**	**497.6**	**6.7**
16	**201**	**87.5**	**341**	**10.7**

Table 3: Speed up of VARSHA with k=8

Figure 6 shows the graph for the communication overhead for various cluster architectures. It may be mentioned that these timings have been obtained over a large sample and in fact, the number of runs made have been over 100 for each case and the variations have been inconsequential.

Scaling of Bandwidth (k)	No. of Boards				
	1	2	4	8	16
1	1.0	1.8	2.7	3.3	3.4
2	1.0	1.9	3.2	4.7	5.6
4	1.0	1.9	3.6	5.9	8.2
8	1.0	2.0	3.8	**6.7**	**10.7**
16	1.0	2.0	3.9	7.2	12.6

Table 4. Bandwidthwise Speedup of original test code

It is not hard to believe that for globally coupled class of problems as the problem increases in size, the bulk of computations increases, and so does the volume of communication; and the disparity gets rather magnified. But the communication timing analysis from the Figure 6 clearly shows that the Floswitch based communication has

performed relatively better as compared to the InfiniBand based communication owing to its low developmental cost.

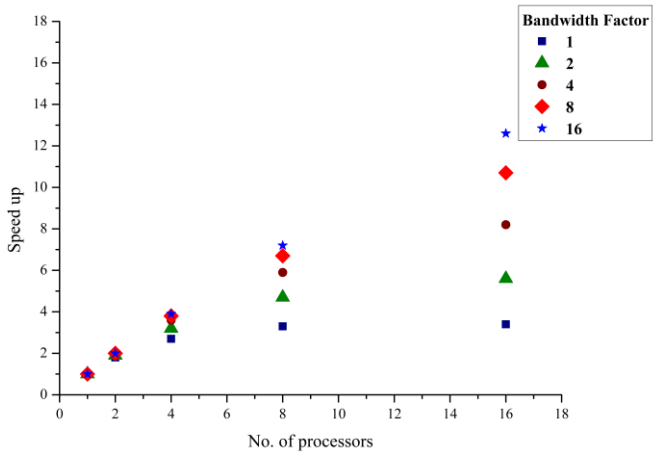

Fig 5. Effect of Bandwidth scaling on speedup.

However, the present work that has involved the reconfiguration of the existing gate arrays, the resultant new protocol has showed an impact of increase in the efficiency by the order of 5 without any additional investment in the hardware. The Floswitch based communication follows "message processing while message passing". It may be highly worthwhile to note that communication timing for the Floswitch is inclusive of the computations carried out in the Floswitch. It may also be seen that the architecture of the Floswitch consists of gate arrays configured to be used for computations where reading of input data and writing of the output data are also inclusive in the protocol. This makes it extremely sensible to comprehend the fact that achievement of improved gain in the new design of Floswitch, directly creates an impact on the communication time of Floswitch which is given in the Table 5. Thus the gain of 5 achieved in the performance of the Floswitch in the present work may be incorporated as the new design of Floswitch

and the timings for it may be seen as shown in Table 6. The Figure 7 shows the graphical representation of the performance of the new design.

Interconnection type used	No. of processors					
	2	**4**	**8**	**16**	**32**	**64**
Ethernet	0.180	0.324	1.034	1.972	3.668	5.467
InfiniBand	0.022	0.032	0.042	0.055	0.066	0.082
Floswitch	0.346	0.572	0.878	1.105	1.495	2.092

Table 5. Timings (in *m*secs) for parallel computations on various no. of processors with different architectures

It may be examined that the new design of the Floswitch has the communication performance comparable with that of the InfiniBand based communication which can be obtained with only the reconfiguration of the existing gate arrays and at no additional investment in the hardware design.

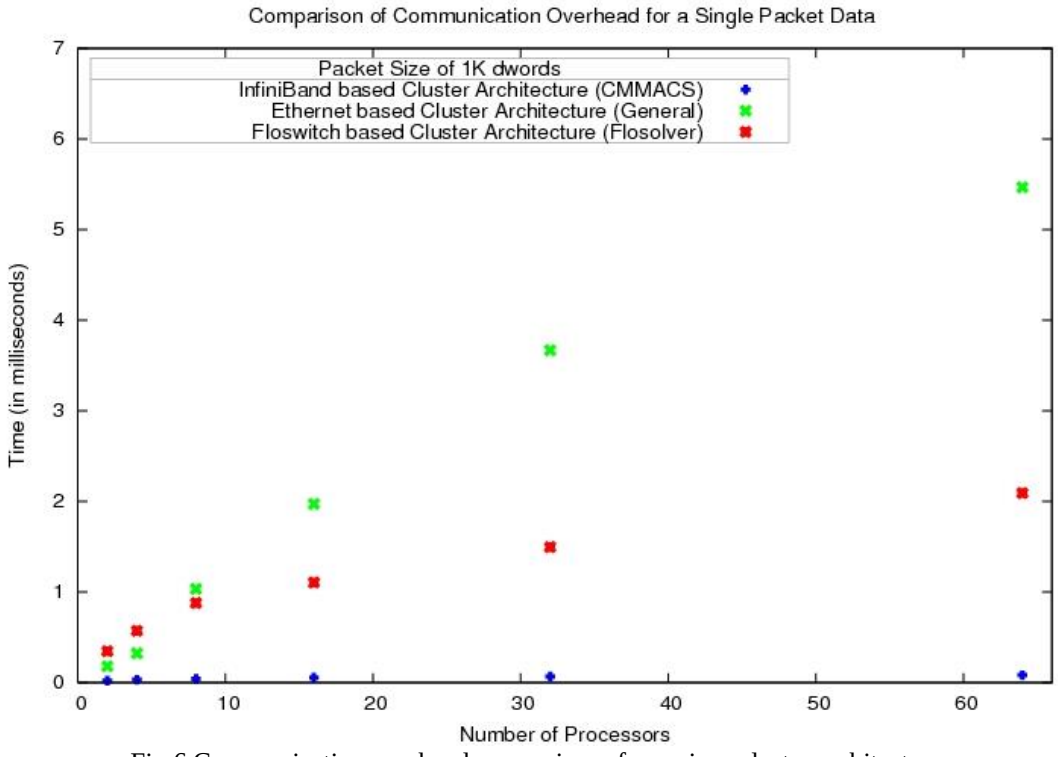

Fig 6 Communication overhead comparisons for various cluster architectures

Interconnection type used	No. of processors					
	2	4	8	16	32	64
Ethernet	0.180	0.324	1.034	1.972	3.668	5.467
InfiniBand	0.022	0.032	0.042	0.055	0.066	0.082
Floswitch	0.346	0.572	0.878	1.105	1.495	2.092
New Design	0.0692	0.1144	0.176	0.221	0.299	0.418

Table 6. Timings obtained for the new design of the Floswitch.

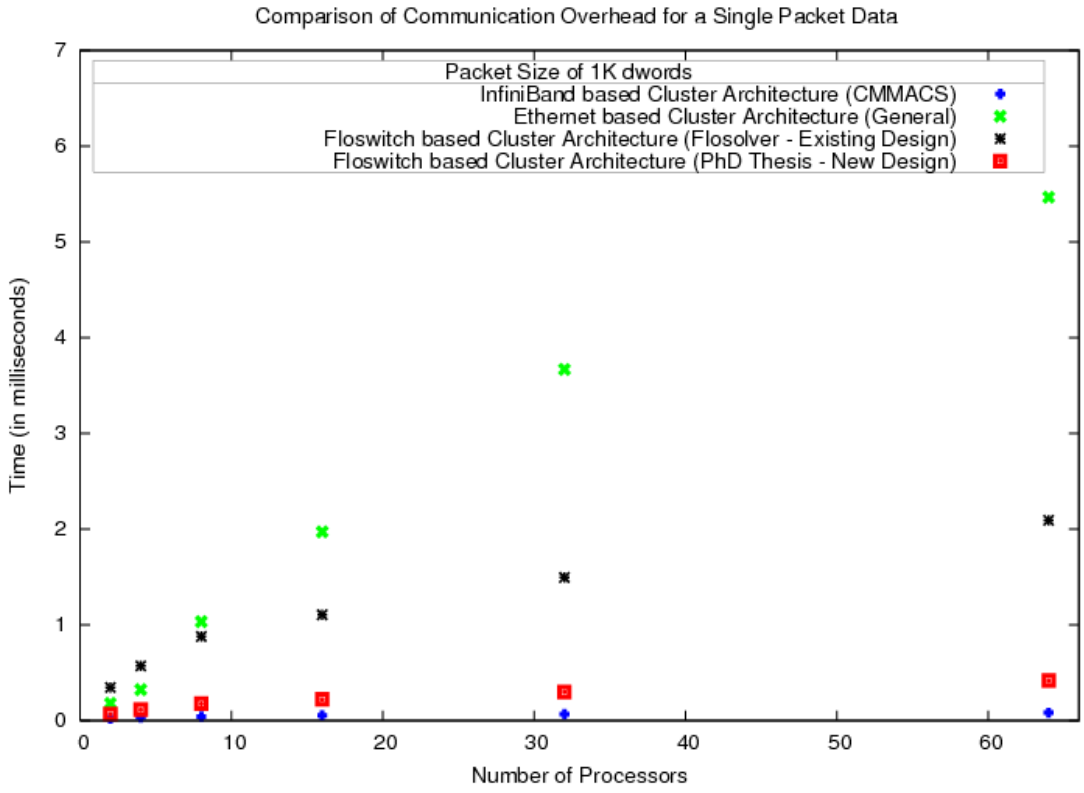

Fig 7. Performance comparison of the new design of the Floswitch.

The present thesis is organized into nine chapters that are briefly described as follows:

Chapter 1 provides the detailed survey of the communication technologies in parallel computers around the world and in India along with the study of their performance index.

Chapter 2 gives the brief account of the historical perspective of scientific parallel computing in relation to the Flosolver programme that eventually gave rise to a new paradigm in communication technology using the Floswitch.

Chapter 3 discusses the parameters which were instrumental in the evolution of Floswitch and its performance with reference to computationally challenging problems like TSP (Transonic Small Perturbation) equations which is non-linear partial differential equations whose type is mixed i.e. in some part it is subsonic and in some part it is supersonic and the transition occurs through sonic line which makes the problem both theoretically and computationally difficult; solution to Navier- stokes equation whose importance in CFD (Computational Fluid Dynamics) is difficult to overstate; Panel code and Direct Numerical Simulation of Navier Stokes Equation that are highly computation intensive and demand global communication.

Chapter 4 summarises the Flosolver series of parallel computers, their distinctiveness in comparison with the contemporary parallel computers with an emphasis on the role of Floswitch and its uniqueness as compared to other state of art communication devices.

Chapter 5 focuses on the scalability issues of Flosolver series of parallel computers and discusses the role of optical communication in its realisation. Existing state of technological development in the Flosolver programme is also briefly described.

Chapter 6 sketches the next stage of logical development which forms the core of the present investigation. The traditional approach of using off the shelf processor for Floswitch characterised in the earlier series of Floswitch is discarded and the fresh approach is suggested which turns out to be efficient by an order of magnitude. A brief discussion presented also includes the data flow mechanism in the new architecture and the stepwise implementation techniques.

Chapter 7 presents the qualitative analysis of the communication times in scientific parallel computing in order to substantiate the performance statistics of the Floswitch compared to the other commercially available communication devices for large scale computing systems.

Chapter 8 analyses the results obtained, and

Chapter 9 concludes the work with a scope for future work.

OBJECTIVES ACCOMPLISHED

A brief study on the different Scientific Parallel Computers around the world with the emphasis on the role of communication technology was performed. The role of Floswitch as the Indian contribution to communication technology for parallel computers was analysed. The Floswitch in Flosolver series of Parallel Computers is the customized communication device that performs computations as well as communication. The computations are performed using the reconfigurable gate arrays and the communication being performed at dual levels using optical communication and the PCI (Peripheral Component Interconnect).

This work addresses the issue of improving the performance of the Floswitch by a factor of 5 without any additional investment in hardware and is achieved by better utilization and synchronization of existing gate arrays and avoiding unnecessary delays in the

circuits. Pipelining became an important tool and as an off suite many new avenues for future development emerged. The improvement evolves into a completely new design of the Floswitch.

In order to put the improvement in the performance of the Floswitch in perspective, a qualitative timing analysis of application code VARSHA was carried out on the Flosolver Mk-8 parallel computer. Figure 4 shows the efficiency of meteorological code VARSHA on a standard cluster based parallel computing system and Figure 5 shows the efficiency of VARSHA code on the same system for different bandwidth scaling factors for different number of processors. The Floswitch based communication follows "message processing while message passing". It may be highly worthwhile to note that communication timing in the Floswitch is inclusive of the computations carried out in the Floswitch. It may also be seen from the previous chapter that the architecture of the Floswitch consists of gate arrays configured to be used for computations where reading of input data and writing of the output data are inclusive in the protocol. Also the new design of the Floswitch showing the performance improvement by a factor of 5, has the communication performance comparable with that of the InfiniBand based communication as shown in Figure 7, which can be obtained with only the reconfiguration of the existing gate arrays and at no additional investment in the hardware design.

As the Floswitch supports optical communication, it is also called as Optical Device and the improvement in its performance is the result of the new efficient algorithms being developed using the existing gate arrays. The research work highlights the significant contribution in the improvement in existing technology that forms the fundamental block of future Large Scale Computing systems.

Mohammed Mahfooz Sheikh
(Research Student)

A M Khan
(Research Supervisor)

REFERENCES

Akira Kasahara, 1970, Computer Simulations of the Global Circulation of the Earths Atmosphere, Computer and their role in the physical sciences, Fernbach S, Taub A, (Eds.), New York, Gordon and Breach Science Publishers, Chapter 23, 571-594.

Bhatkar V P, 1991, Param Architecture, Programming Environment and Applications, 3rd Int. Conf. on Applications of Transputers, Aug 28-30, Glasgow, Scotland, 48 – 59.

Charles Eames, Ray Eames, 1973, A Computer Perspective, Glen Fleck (Ed.), Cambridge, Massachusetts, Harvard University Press.

David Kuck J, 1978, The structure of Computers and Computation, New York, John Wiley & Sons Inc., Vol 1.

Fermi E, Pasta J R, Tsingou M, Ulam S, 1955, Studies of non- linear problems I, Technical Report LA-1940, Los Alamos Scientific Laboratory, Los Alamos, NM, USA.

Hausheer F H, 1992, Numerical simulation, parallel clusters, and the design of novel pharmaceutical agents for cancer treatment, In *Proceedings of the 1992 ACM/IEEE conference on Supercomputing* (Supercomputing '92), Robert Werner (Ed.), IEEE Computer Society Press, Los Alamitos, CA, USA, 636-637.

Herman Goldstine H, John Von Nuemann, 1976, Blast Wave Calculation, John Von Neumann collected works – Theory of games, Astrophysics, Hydrodynamics and Meteorology, Taub A H (Ed.), Oxford, Pergamon Press Ltd., Article 29, VI, 386 – 412.

Herman Goldstine H, 1973, The Computer: from Pascal to Von – Neumann, Second Edition, Princeton, New Jersey, Princeton University Press.

Jon Squire S, Sandra Palais M, 1963, Programming and design considerations of a highly parallel computer, In *Proceedings of the May 21-23, 1963, spring joint computer conference* (AFIPS '63 (Spring)). ACM, New York, NY, USA, 395-400.

John Mauchly W, 1980, The ENIAC, A History of Computing in the Twentieth Century, Metropolis N, Howlett J, Gian Carlo Rotta (Eds.), New York, Academic Press, 541 – 550.

Koczela L J, Wang G Y, 1969, The Design of a Highly Parallel Computer Organization, *IEEE Trans. Comput.* 18(6), 520-529.

Mark Delderfield, Lee Kitching, Gareth Smith, David Hoyle, Iain Buchan, 2008, Shared Genomics: Accessible High Performance Computing for Genomic Medical Research, In *Proceedings of the 2008 Fourth IEEE International Conference on eScience* (ESCIENCE '08). IEEE Computer Society, Washington, DC, USA, 404-405.

Michael Quinn J, 1988, Designing Efficient Algorithms for Parallel Computers, Second Reprint, McGraw Hill International Edition.

Neelakantan K, Ghosh P P, Ganagi M S, Athithan G, Atre M V, Venkataraman G, 1990, Performance characteristics of a hypercube-type parallel computer, Current Science, 59(20), 982 – 988.

Presper Eckart Jr J, 1980, The ENIAC, A History of Computing in the Twentieth Century, Metropolis N, Howlett J, Gian Carlo Rotta (Eds.), New York, Academic Press, 525 – 540.

Sinha U N, Deshpande M D, Sarasamma V R, 1988, Flosolver: A Parallel Computer for Fluid Dynamics, Current Science, 57(23), 1277 – 1285.

Venkatesh T N, Sarasamma V R., Rajalakshmy S, Kirti Chandra Sahu, Rama Govindarajan, 2005, Super-linear speed-up of a parallel multigrid Navier – Stokes solver on Flosolver, Current Science, 88(4), 589 – 593.

PAPERS PUBLISHED/SUBMITED

Journal:

1. Mohammed Mahfooz Sheikh, A M Khan, T Venkatesh and U N Sinha, "The Assessment of Bandwidth Requirements for Meteorological Code VARSHA on a Parallel Computing System", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 4, July-August 2012, pp.1261-1268.
2. Mohammed Mahfooz Sheikh, A M Khan, T Venkatesh and U N Sinha, "On the Timing Analysis of Cluster based Communication devices for Large Scale Computing Systems", International Journal of Computer Science and Engineering (IJCSE). (Accepted for Publication)
3. Mohammed Mahfooz Sheikh, A M Khan, R P Thangavelu and U N sinha, "On Scalability Issues of Cluster Computing", Indian Journal of Geo-Marine Sciences. (Submitted)
4. Mohammed Mahfooz Sheikh, A M Khan, T Venkatesh and U N Sinha, "On Issues Related to Parallelisation Strategy of Large Application Codes - A Case Study", Parallel Processing Letters. (Submitted)

Conferences:

1. Mohammed Mahfooz Sheikh, A M Khan, T Venkatesh and U N Sinha, "Communication and Computation Timing Analyses of Large Scale Computing Systems – A Case Study", Proc. of National Conference on Advanced Computing

and Communications (NCACC' 12), 27th – 28th April 2012, pp. 155-161. (Presented & Published in the proceedings)

2. Mohammed Mahfooz Sheikh, A M Khan, T Venkatesh and U N Sinha, "Role of Ethernet and Infiniband Communications in the Cluster based Architecture for Large Scale Computing", Proc. of National Conference on Recent Advances in Electronics and Communication Technology (NCAECT2012), 18th May 2012. (Presented & Published in the proceedings).
3. A M Khan, Mohammed Mahfooz Sheikh and U N Sinha "Cluster Computing – Is it Scalable for Disaster Management?", Intl. Conf. on Electronic Design and Signal Processing ICEDSP'12, December 2012, MIT, Manipal. (Accepted for Oral Presentation).
4. A M Khan, Mohammed Mahfooz Sheikh and U N Sinha "The Performance of customized Communication Devices in Cluster based Computing systems", International Conference on Information Technology Convergence and Services (ITCSE 2012), October 2012, Avinashlingam University, Coimbatore. (Communicated)